



Deprecations and removals in 1.3

This PDF is brought to you by
SENSIOLABS 

License: Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 Unported License
Version: deprecated-1.4-en-2009-12-05

Table of Contents

Array

Deprecations and removals in 1.3

This document lists all settings, classes, methods, functions, and tasks that have been deprecated or removed in symfony 1.3.

Core Plugins

The following core plugins have been deprecated in symfony 1.3 and will be removed in symfony 1.4:

- `sfCompat10Plugin`: By deprecating this plugin, we also deprecate all other elements in the framework that rely on this plugin to work (1.0 admin generator, and 1.0 form system). It also includes the default admin generator theme for 1.0 located in `lib/plugins/sfPropelPlugin/data/generator/sfPropelAdmin`.
- `sfProtoculousPlugin`: The helpers provided by this plugin do not support unobstrusiveness, and as such should not be used anymore.

Methods and Functions

The following methods and functions have been deprecated in symfony 1.3 or before, and will be removed in symfony 1.4:

- `sfToolkit::getTmpDir()`: You can replace all occurrences of this method by `sys_get_temp_dir()`
- `sfToolkit::removeArrayValueForPath()`, `sfToolkit::hasArrayValueForPath()`, and `sfToolkit::getArrayValueForPathByRef()`
- `sfValidatorBase::setInvalidMessage()`: You can replace it by a call to the new `sfValidatorBase::setDefaultMessage()` method
- `sfValidatorBase::setRequiredMessage()`: You can replace it by a call to the new `sfValidatorBase::setDefaultMessage()` method
- `sfTesterResponse::contains()`: You can use the more powerful `matches()` method
- `sfTestFunctionalBase` following methods: `isRedirected()`, `isStatusCode()`, `responseContains()`, `isRequestParameter()`, `isResponseHeader()`, `isUserCulture()`, `isRequestFormat()`, and `checkResponseElement()`: These methods have been deprecated since 1.2, and replaced with the tester classes.
- `sfTestFunctional` following methods: `isCached()`, `isUriCached()`: These methods have been deprecated since 1.2, and replaced with the tester classes.

- `sfFilesystem::sh()`: You can replace all occurrences of this method by calls to the new `sfFilesystem::execute()` method. Be warned that the returned value of this method is an array composed of the `stdout` output and the `stderr` output.
- `sfAction::getDefaultView()`, `sfAction::handleError()`, `sfAction::validate()`: These methods have been deprecated in symfony 1.1, and they were not really useful. As of symfony 1.1, they need the `compat_10` setting set to `on` to work.
- `sfComponent::debugMessage()`: Use the `log_message()` helper instead.
- `sfApplicationConfiguration::loadPluginConfig()`: Use `initializePlugins()` instead.
- `sfLoader::getHelperDirs()` and `sfLoader::loadHelpers()`: Use the same methods from the `sfApplicationConfiguration` object. As all methods of the class `sfLoader` are deprecated, the `sfLoader` class will be removed in symfony 1.4.
- `sfController::sendEmail()`: Use the new mailer feature of Symfony 1.3 instead.
- `sfGeneratorManager::initialize()`: It does nothing.
- `debug_message()`: Use the `log_message()` helper instead.
- `sfWebRequest::getMethodName()`: Use `getMethod()` instead.
- `sfDomCssSelector::getTexts()`: Use `matchAll()->getValues()`
- `sfDomCssSelector::getElements()`: Use `matchAll()`
- `sfVarLogger::getXDebugStack()`: Use `sfVarLogger::getDebugBacktrace()` instead.
- `sfVarLogger`: The logged `debug_stack` value is deprecated in favor of the `debug_backtrace` value.
- `sfContext::retrieveObjects()`: The method is only used by `ObjectHelper`, which is deprecated

The following methods and functions have been removed in symfony 1.3:

- `sfApplicationConfiguration::checkSymfonyVersion()`: see below for the explanation (`check_symfony_version` setting)

Classes

The following classes have been deprecated in symfony 1.3 and will be removed in symfony 1.4:

- `sfDoctrineLogger`: Use `sfDoctrineConnectionProfiler` instead.
- `sfNoRouting` and `sfPathInfoRouting`
- `sfRichTextEditor`, `sfRichTextEditorFCK`, and `sfRichTextEditorTinyMCE`: They have been replaced by the widget system (see the “Helpers” section below)
- `sfCrudGenerator`, `sfAdminGenerator`, `sfPropelCrudGenerator`, `sfPropelAdminGenerator`: These classes were used by the 1.0 admin generator
- `sfPropelUniqueValidator`, `sfDoctrineUniqueValidator`: These classes were used by the 1.0 form system
- `sfLoader`: see the “Methods and Functions” section
- `sfConsoleRequest`, `sfConsoleResponse`, `sfConsoleController`

- `sfDoctrineDataRetriever`, `sfPropelDataRetriever`: These classes are only used by `ObjectHelper`, which is deprecated
- `sfWidgetFormI18nSelectLanguage`, `sfWidgetFormI18nSelectCurrency`, and `sfWidgetFormI18nSelectCountry`: Use the corresponding `Choice` widgets (`sfWidgetFormI18nChoiceLanguage`, `sfWidgetFormI18nChoiceCurrency`, and `sfWidgetFormI18nChoiceCountry` respectively) as they act exactly in the same way, except they have more customization possibilities
- `sfWidgetFormChoiceMany`, `sfWidgetFormPropelChoiceMany`, `sfWidgetFormDoctrineChoiceMany`, `sfValidatorChoiceMany`, `sfValidatorPropelChoiceMany`, `sfValidatorPropelDoctrineMany`: Use the same classes but without `Many` at the end, and set the `multiple` option to `true`
- `SfExtensionObjectBuilder`, `SfExtensionPeerBuilder`, `SfMultiExtendObjectBuilder`, `SfNestedSetBuilder`, `SfNestedSetPeerBuilder`, `SfObjectBuilder`, `SfPeerBuilder`: The custom `Propel` builder classes have been ported to `Propel 1.4`'s new behaviors system

The following classes have been deprecated in symfony 1.3:

- `sfCommonFilter`: see the “Removal of the common filter” section of the `UPGRADE_TO_1_3` file for more information about the consequences and how to migrate your code.

Helpers

The following helper groups have been deprecated in symfony 1.3 and will be removed in symfony 1.4:

- All helpers related to the 1.0 form system as provided by the `sfCompat10Plugin`: `DateForm`, `Form`, `ObjectAdmin`, `Object`, and `Validation`

The `form_tag()` helper from the `Form` helper group has been moved to the `Url` helper group, and as such is still available in symfony 1.4.

Loading helpers from the `PHP` include path has been deprecated in 1.3 and removed in 1.4. Helpers must be located in one of the `project`, `application` or `module lib/helper/` directories.

Settings

The following settings (managed in the `settings.yml` configuration file) have been removed from symfony 1.3:

- `check_symfony_version`: This setting was introduced years ago to allow automatic cache cleaning in case of a change of the symfony version. It was mainly useful for shared hosting configuration where the symfony version is shared amongst all customers. As this is bad practice since symfony 1.1 (you need to embed the symfony version in each of your project), the settings does not make sense anymore. Moreover, when the setting is set to `on`, the check adds a small overhead to each request, as we need to get the content of a file.
- `max_forwards`: This settings controls the number of forwards allowed before symfony throws an exception. Making it configurable has no value. If you need more than 5 forwards, you have both a conception problem and a performance one.

- `sf_lazy_cache_key`: Introduced as a big performance improvement in symfony 1.2.6, this setting allowed you to turn on a lazy cache key generation for the view cache. While we think doing it lazy was the best idea, some people might have relied on `SfViewCacheManager::isCacheable()` being called even when the action itself wasn't cacheable. As of symfony 1.3, the behavior is the same as if `sf_lazy_cache_key` was set to `true`.
- `strip_comments`: The `strip_comments` was introduced to be able to disable the comment stripping because of some bugs in the tokenizer of some PHP 5.0.X versions. It was also used later on to avoid large memory consumption when the Tokenizer extension was not compiled with PHP. The first problem is not relevant anymore as the minimum version of PHP needed is 5.2 and the second one has already been fixed by removing the regular expression that simulated the comment stripping.
- `lazy_routes_deserialize`: This option is not needed anymore.

The following settings have been deprecated in symfony 1.3 and will be removed in symfony 1.4:

- `calendar_web_dir`, `rich_text_js_dir`: These settings are used by the Form helper group, which is deprecated in symfony 1.3.
- `validation_error_prefix`, `validation_error_suffix`, `validation_error_class`, `validation_error_id_prefix`: These settings are used by the Validation helper group, which is deprecated in symfony 1.3.
- `is_internal` (in `module.yml`): The `is_internal` flag was used to prevent actions from being called from a browser. This was added to protect email sending in symfony 1.0. As email support does not require this trick anymore, this flag will be removed and not checked anymore in the symfony core code.

Tasks

The following tasks have been removed in symfony 1.3:

- `project:freeze` and `project:unfreeze`: These tasks used to embed the symfony version used by a project inside the project itself. They are not needed anymore as the best practice has been to embed symfony in the project for a very long time now. Moreover, switching from one version of symfony to another is really simple now as you only need to change the path in the `ProjectConfiguration` class. Embedding by hand symfony is also very simple as you just need to copy the whole symfony directory somewhere in your project (`lib/vendor/symfony/` is the recommended one).

The following tasks are deprecated in symfony 1.3, and will be removed in symfony 1.4:

- All symfony 1.0 task aliases.
- `propel:init-admin`: This task generated admin generator modules for symfony 1.0.

The following Doctrine tasks have been merged into `doctrine:build` and will be removed in symfony 1.4:

- `doctrine:build-all`
- `doctrine:build-all-load`
- `doctrine:build-all-reload`
- `doctrine:build-all-reload-test-all`

- doctrine:rebuild-db
- doctrine:reload-data

Miscellaneous

The following behaviors are deprecated in symfony 1.3, and will be removed in symfony 1.4:

- The `sfParameterHolder::get()`, `sfParameterHolder::has()`, `sfParameterHolder::remove()`, `sfNamespacedParameterHolder::get()`, `sfNamespacedParameterHolder::has()`, and `sfNamespacedParameterHolder::remove()` methods support for the array notation (`[]`) is deprecated and won't be available in symfony 1.4 (better for performance).

The symfony CLI does not accept anymore the global `--dry-run` option as it was not used by any symfony built-in task. If one of your task relies on this option, you can just add it as a local option of your task class.

The Propel templates for the 1.0 admin generator and the 1.0 CRUD will be removed in symfony 1.4 (`plugins/sfPropelPlugin/data/generator/sfPropelAdmin/`).

The “Dynarch calendar” (found in `data/web/calendar/`) will be removed in symfony 1.4 as it is only used by the Form helper group, which will be also removed in symfony 1.4.

As of symfony 1.3, the `unavailable` page will only be looked for in the `%SF_APP_CONFIG_DIR%/` and `%SF_CONFIG_DIR%/` directories. If you still have it stored in `%SF_WEB_DIR%/errors/`, you must move it before migrating to symfony 1.4.

The `doc/` directory at the root of a project is not generated anymore, as it was not used by symfony itself. And so the related `sf_doc_dir` has also been removed.

The `sfDoctrinePlugin_doctrine_lib_path` setting, previously used to specify a custom Doctrine lib directory, has been deprecated in 1.3 and removed in 1.4. Use the `sf_doctrine_dir` setting instead.

All symfony Base* classes generated classes are not marked as `abstract`.